

Session Code: ARC413
**architecture
& infrastructure**

“Whidbey” CLR Internals

Jonathan Keljo
jkeljo@microsoft.com
Program Manager
Microsoft Corporation

PDC⁰³

Make the connection

Microsoft[®]

WinFX Developer Preview

Tools

Visual Studio .net
Visual Basic .net
Visual C++ .net
Visual C# .net
Visual J# .net

Client Application Model

Avalon

System.Windows

Windows Forms

System.Windows.Forms

Web & Service Application Model

ASP.NET / Indigo

System.Web

Data Systems Application Model

Win FS

System.Storage

Yukon

System.Data.SqlServer

Mobile PC & Devices Application Model

Compact Framework

System.Windows.Forms

Mobile PC Optimized

System.Windows

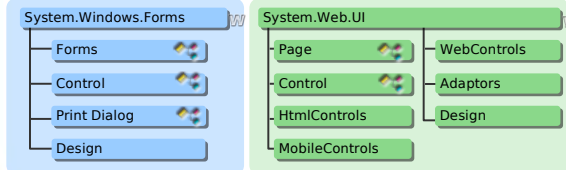
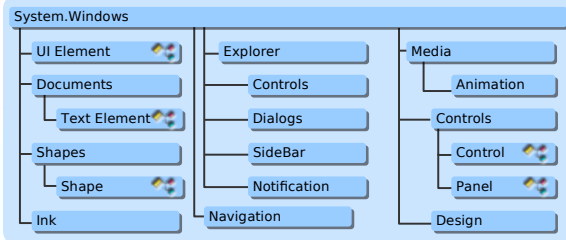
Command Line

System.Console

NT Service

System.ServiceProcess

Presentation



System.Help

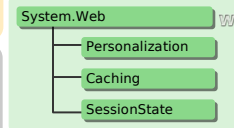
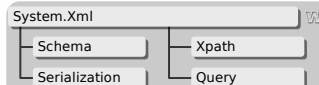
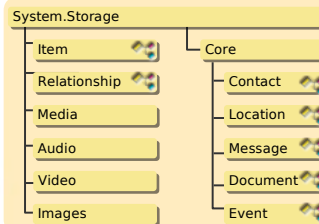
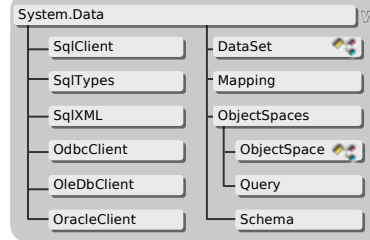
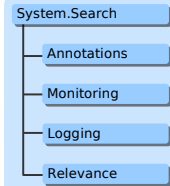
System.Drawing

System.NaturalLanguageServices

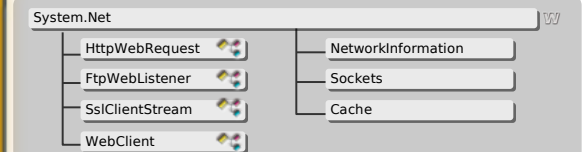
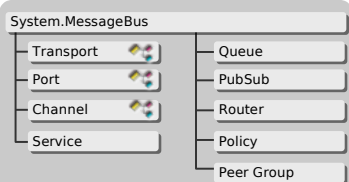
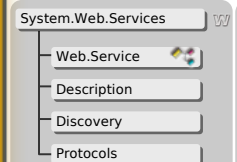
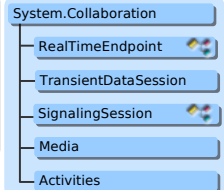
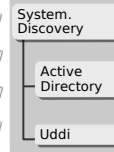
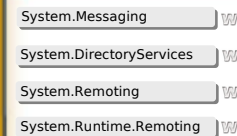
System.Speech

- Recognition
- Synthesis

Data

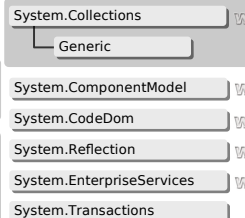
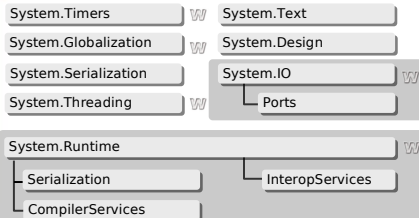


Communication

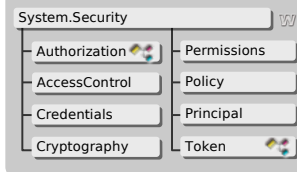
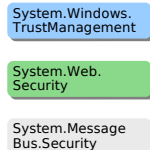


Fundamentals

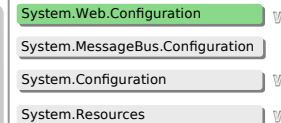
Base & Application Services



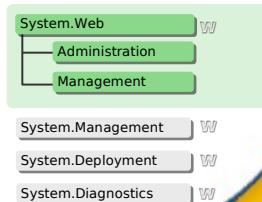
Security



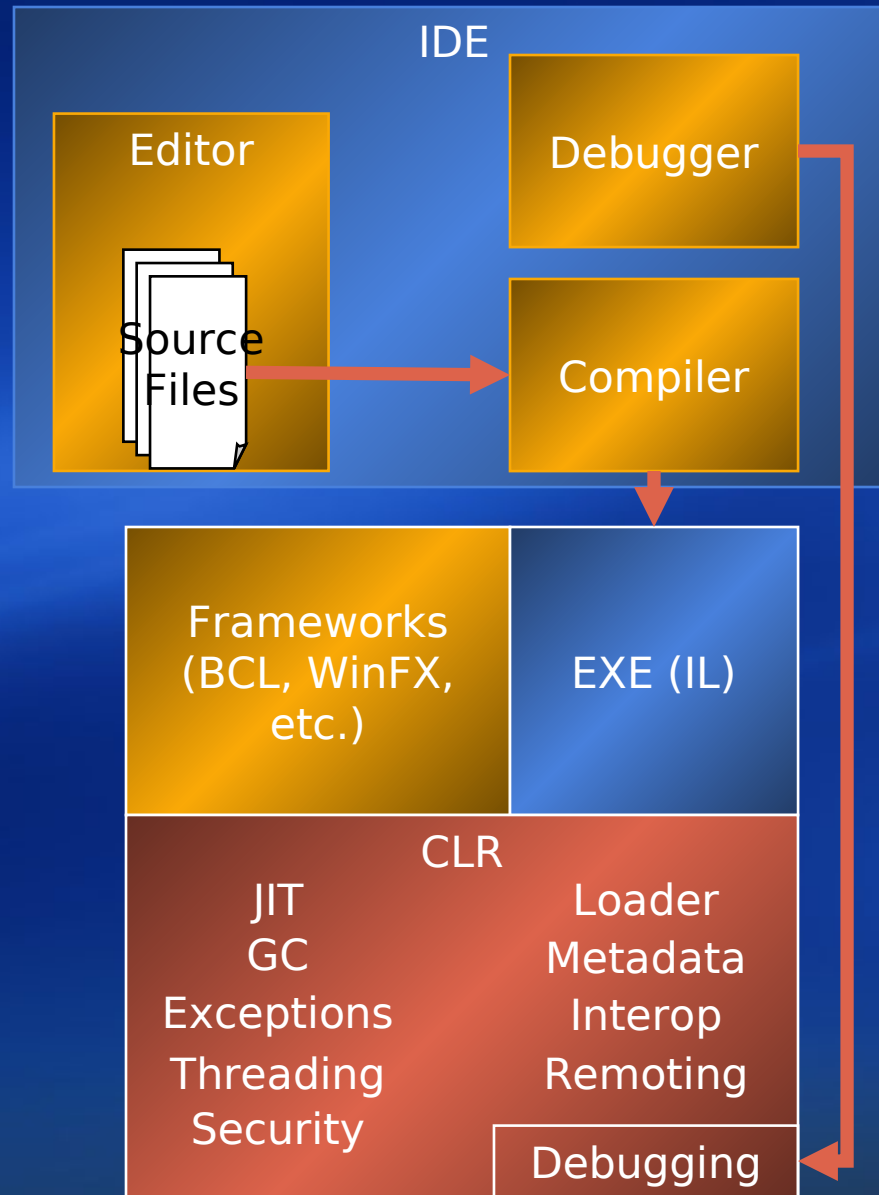
Configuration



Deployment/Management



The Big Picture



C++ Templates

```
template <class T>
class Stack
{
    ...
    T Pop() ...
}
```

Template stack
class

```
Stack<int> ...
...
Stack <double> ...
...
Stack <point> ...
...
```

Main program
source

C++
Compiler

Stack
<int>

Stack
<double>

Stack
<point>

⋮

Main
program

Executable (x86)

CLR Generics

```
class Stack<T>  
{  
    ...  
    T Pop() ...  
}
```

Template stack
class

```
Stack<int> ...  
...  
Stack <double> ...  
...  
Stack <point> ...  
...
```

Main program
source

.NET
Compiler

Stack

<T>

Main
program

Executable (IL)

Generics in IL & Metadata

demo

PDC⁰³

Make the connection

IL Execution

IL (Source line "T retval = _array[topOfStack]")

ldarg.0

ldfld !0[] class GenericsDemo.GenericStack<!

0>::_array

ldloc.0

ldelem !0

stloc.1

Argument
s

[0]	this
-----	------

Locals

[0]	topOfStack
-----	------------

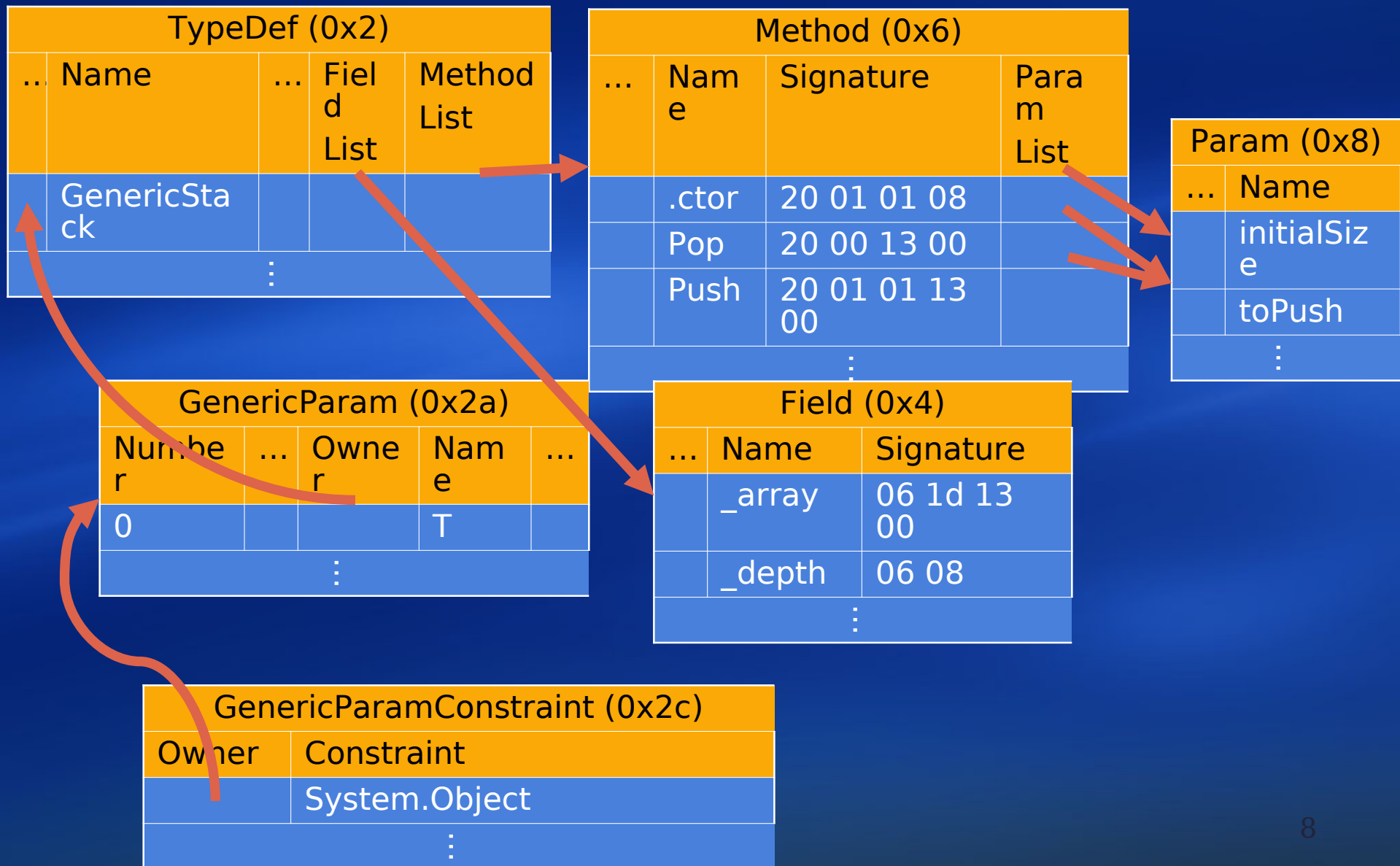
[1]	this._array[topOfStack]
-----	-------------------------

topOfStack

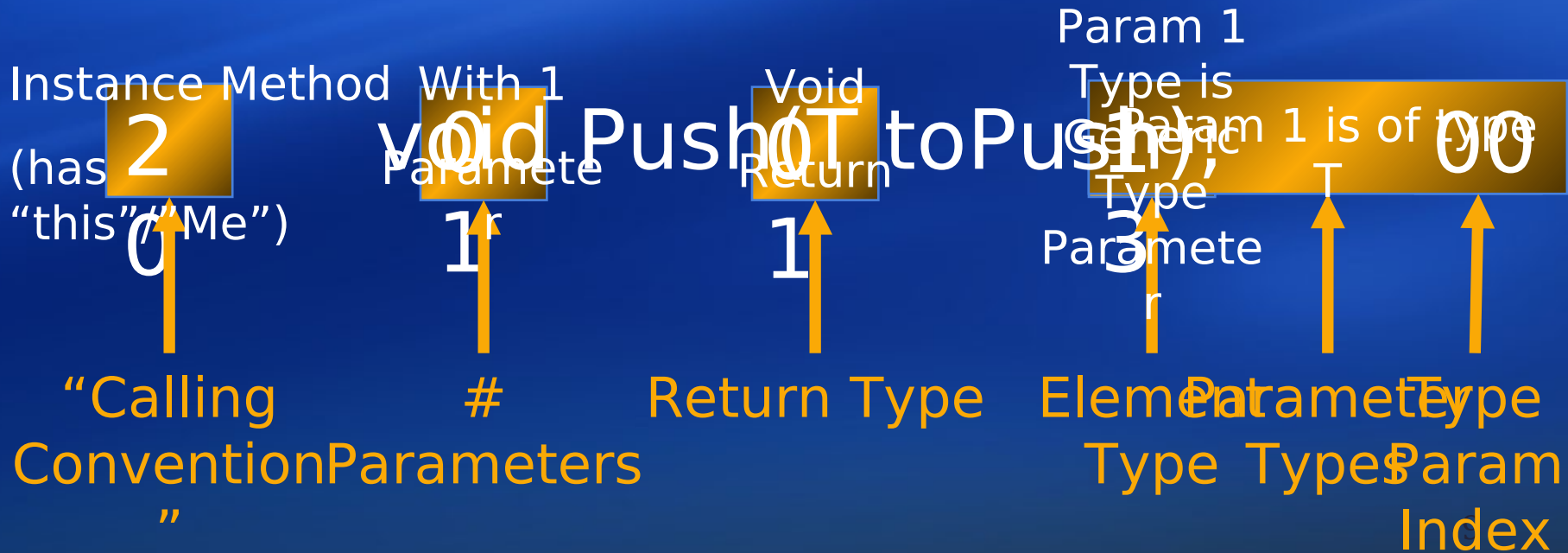
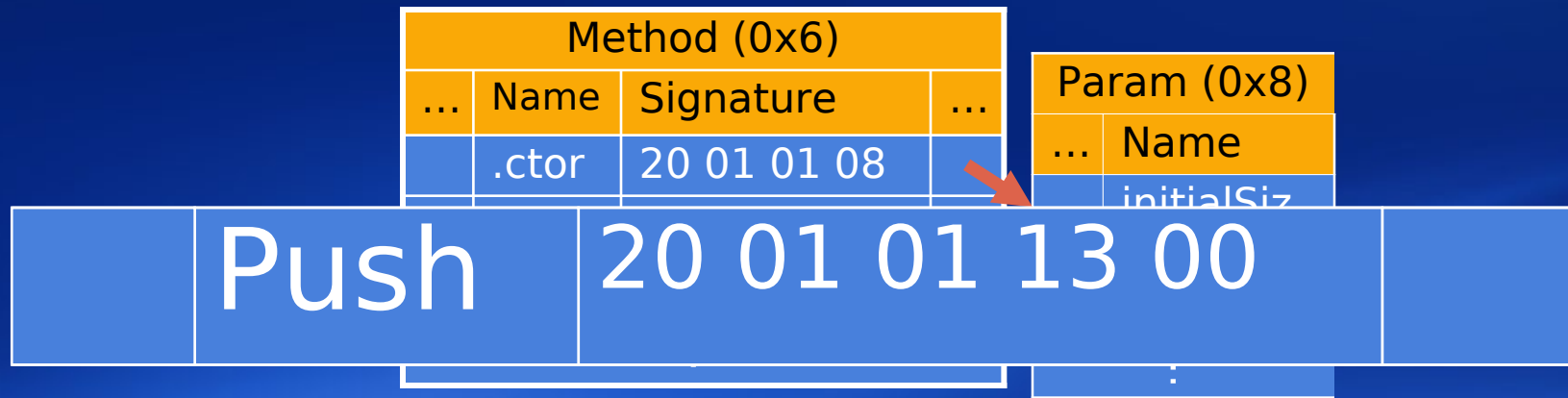
this._array[topOfStack]

Stack

Generics in Metadata



Generic Signatures



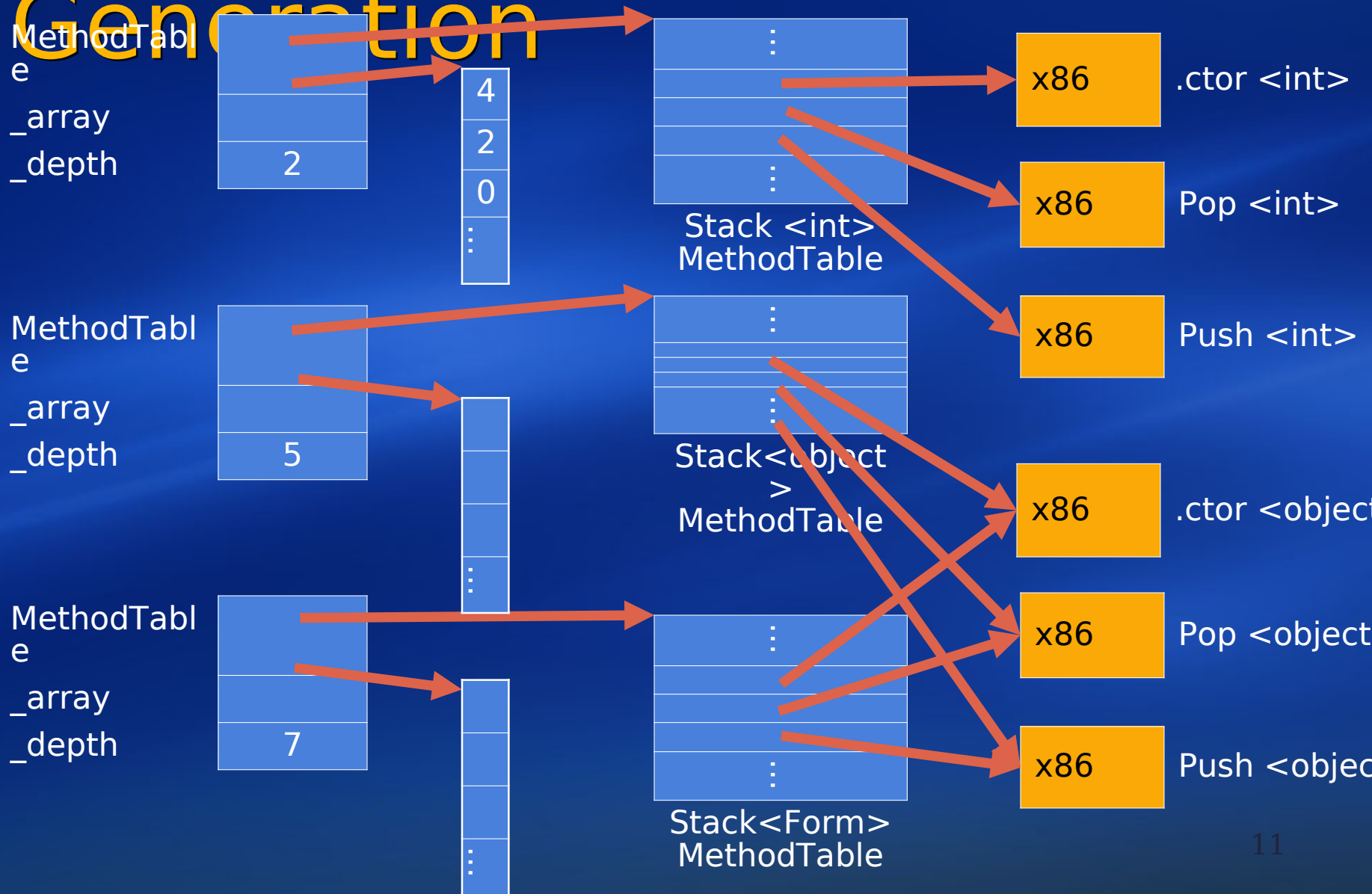
Generics: Code Generation

demo

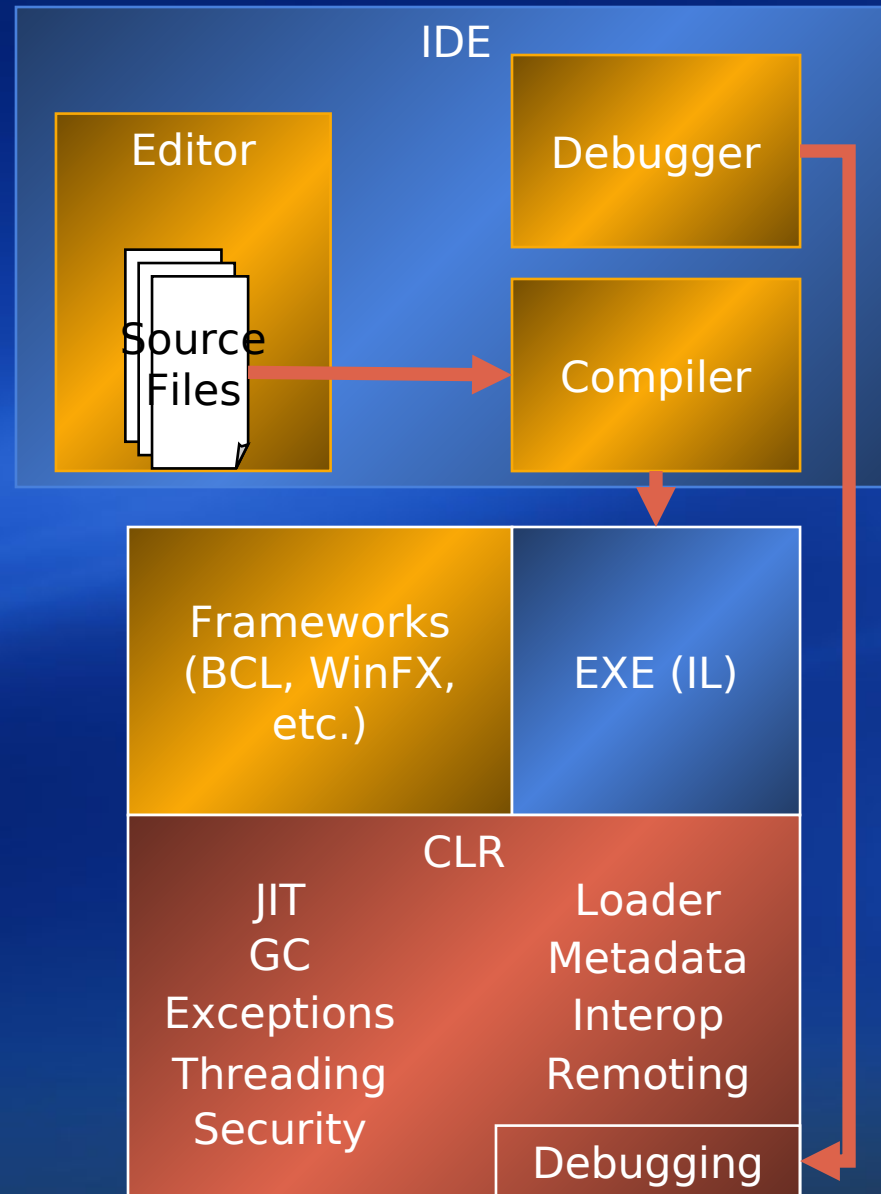
PDC⁰³

Make the connection

Generics - Code Generation



The Big Picture



EnC: Editing a Method

demo

PDC⁰³

Make the connection

EnC – Editing a Method (1)

Delta IL & Delta Metadata

Delta Metadata (Provided by debugger)

Method (0x6)			
RV	...	Name	...
A		PrintStatistic	
...		...	

EnCLog (0x1e)	
Token	FuncCode
06000002	0
...	...

EnCMap (0x1f)	
Token	
06000002	
...	...

Delta IL

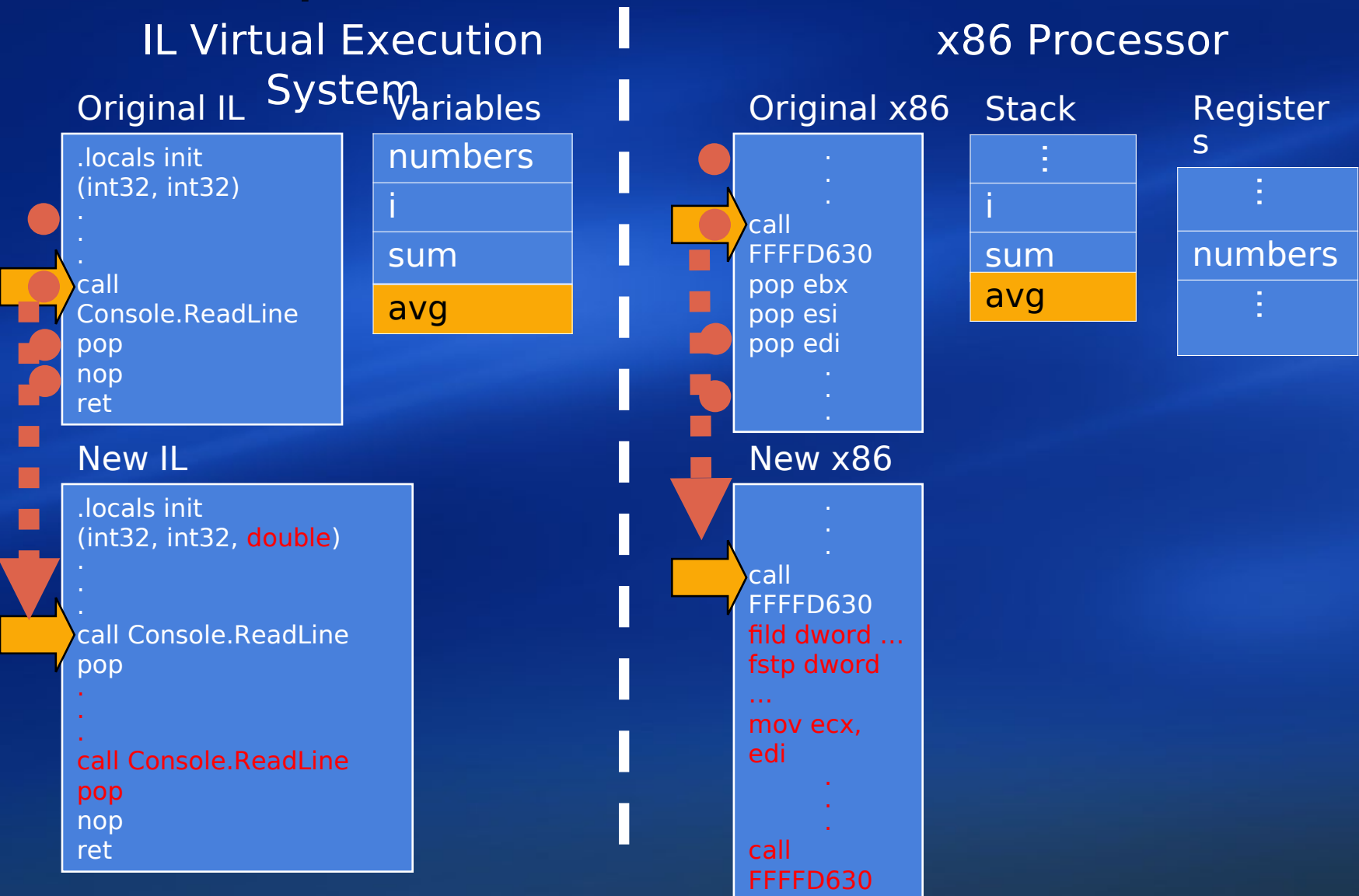
0	Delta Header
4	IL Header for PrintStatistics
	nop ldc.i4.0 stloc.1 ldc.i4.0 : :

Full Metadata (In Running Process)

Method (0x6)			
RVA	...	Name	...
205		Main	
INV		PrintStatistic	
20a8		c805	
		0	
		20a8	
		Complete Original IL	
		Other Data	
		Delta IL	

EnC – Editing a Method (2)

Remap



EnC – Editing Within a Call

IL Virtual Execution System

Original IL

```
.locals init  
(int32, int32)  
:  
:  
:  
call Console.ReadLine  
pop  
nop  
ret
```

Variables

numbers

i

sum

avg

New IL

```
.locals init  
(int32, int32, double)  
:  
:  
:  
call Console.ReadLine  
pop  
:  
:  
call Console.ReadLine  
pop  
nop  
ret
```

x86 Processor

Original x86

```
call  
FFFD630  
pop ebx  
pop esi  
pop edi  
:  
:
```

Stack

```
:  
i  
sum  
vars for  
ReadLine  
:  
:  
:
```

Registers

```
numbers  
:  
:
```

New x86

```
call  
FFFD630  
fld dword ...  
fstp dword ...  
mov ecx, edi  
:  
:  
call  
FFFD630
```

EnC: Editing a Class

demo

EnC – Editing a Class (1)

Delta Metadata

Delta Metadata (Provided by

debugger)

Field (0x4)		
...	Name	Signature
	_avg	06 0d
	_max	06 08
	⋮	

EnCLog (0x1e)	
Token	Func Code
02000001	2
04000019	0
02000001	2
04000020	0
⋮	

EnCMap (0x1f)	
Token	
04000019	
04000020	
⋮	

Full Metadata (In Running Process)

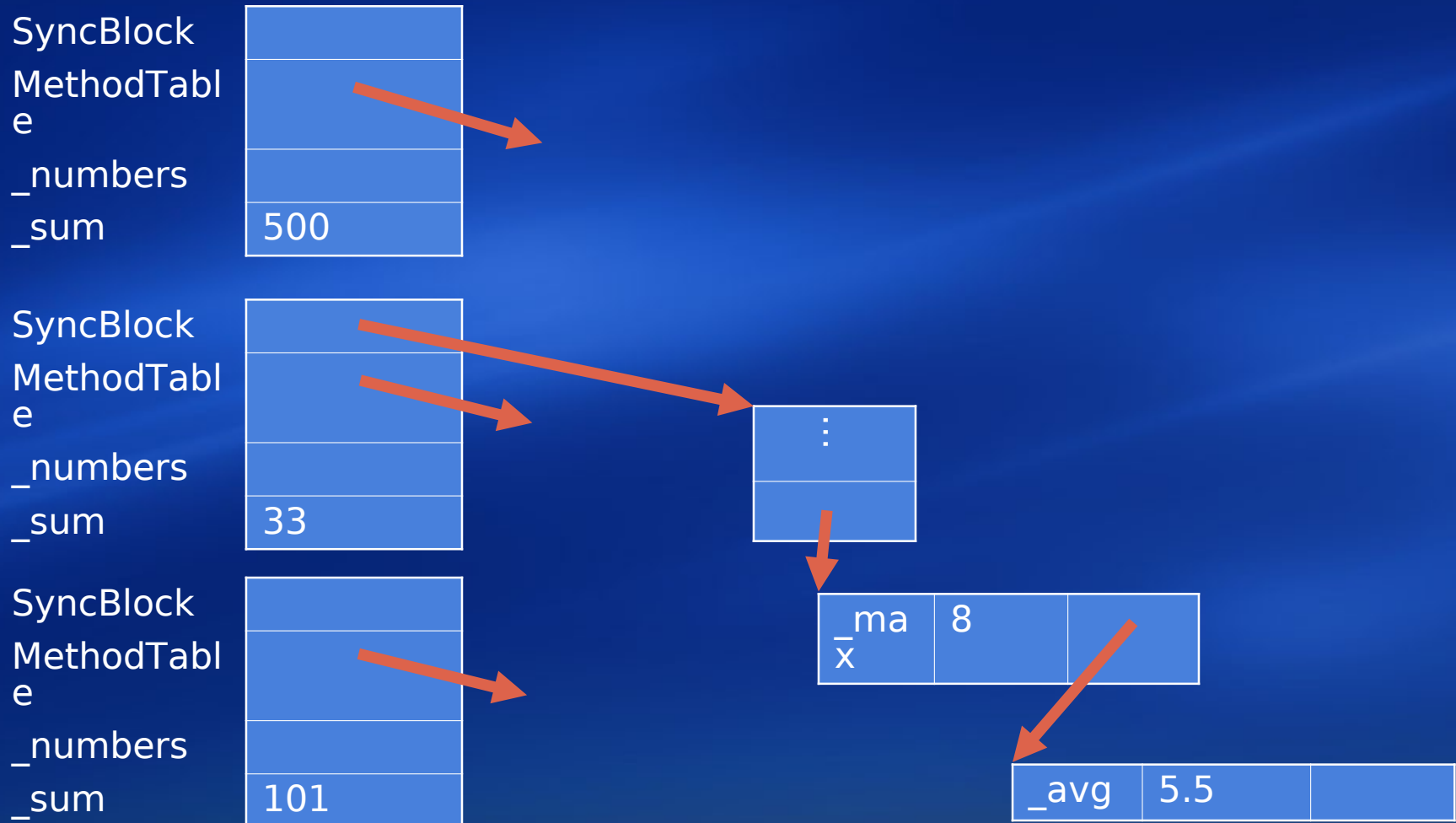
TypeDef (0x2)			
..	Name	...	Field List
	StatBuilder		030000001
	⋮		

FieldPtr (0X3)	
Token	
04000001	
04000002	
04000019	
04000020	
⋮	

Field (0x4)		
...	Name	Signature
	_number	06 12 45
	_s	
	_sum	06 08
	⋮	
	_avg	06 0d
	_max	06 08

EnC – Editing a Class (2)

Extending Existing Instances



Keeping Things Sane

Runtime Power vs. Language Legality

- Can do all kinds of crazy things with CLR's support
- Left to the languages to decide what edits make sense
 - Lots of potential work here
 - EnC-friendly codegen
 - Exception handling
 - Managing cascading effects of edits

Shameless Plug

- Language Implementers' Lunch
 - Thursday 11:30-1:30
 - Programming Languages & Tools Track Lounge

Community Resources

Get Your Questions Answered!

- Newsgroups:
 - microsoft.private.whidbey.clr
- Architecture & Infrastructure Lounge: 309 Foyer
 - connect with Microsoft infrastructure product teams, and PDC 2003 Speakers
- PDC Weblogs:
 - <http://pdcbloggers.net>

Resources

• Books

- The Common Language Infrastructure Annotated Standard. ISBN 0-3211-5493-2
- Inside Microsoft .NET IL Assembler. ISBN 0-7356-1547-0
- Essential .NET Volume 1. ISBN 0-201-73411-7
- Shared Source CLI Essentials. ISBN 0-596-00351-X

Questions?



PDC⁰³

Make the connection

Microsoft Professional Developers Conference 2003

October 26 - 30, 2003, Los Angeles, CA

Microsoft®